Молдавский государственный университет

Факультет математики и информатики

Департамент Информатики

**Дипломная работа**

Разработка Веб-приложений на основе

Микросервисной Архитектуры

**444.1 Информатика**

Выполнена студентом III курса,

специальности Информатика

**Чобану Артёмом, группа I1902**

Руководитель, преподаватель

Департамента Информатики

**Георгий Латул**

Кишинэу – 2022

Оглавление

[Ключевые слова и сокращения 3](#_Toc93563377)

[Введение 4](#_Toc93563378)

[Глава 1 – RESTful API веб-сервисы. 4](#_Toc93563379)

[Глава 2 – Проработка проекта. 4](#_Toc93563380)

[Глава 3 – Реализация проекта. 4](#_Toc93563381)

[Заключение 4](#_Toc93563382)

[Литература и документация: 5](#_Toc93563383)

# Ключевые слова и сокращения

Веб-сервис, Сервис-Ориентированная Архитектура, Микросервис, Микросервисная Архитектура, Vertical Slice Архитектура, Message Queuing Service (Служба очереди сообщений), API Gateway (API Шлюз), Монолит.

CQRS – Command Query Responsibility Segregation

MQS – Message Queueing Service

SOA – Service Oriented Architecture

API – Application Programming Interface

# Введение

Современные веб-приложения становятся всё сложнее, требования растут, и касаются множества аспектов, базирующихся на предыдущем опыте, и необходимости для конкретного проекта. Ниже приведены основные требования к комплексным веб-приложениям:

1. Отказоустойчивость -

…

1. Простота развёртывания –

…

1. Масштабирование –

…

1. Заменяемость –

…

1. Простота тестирования –

…

1. Количество зависимостей –

…

1. Безопасность –

…

# Глава 1 – Микросервисная Архитектура.

Веб-Сервис – это …

Сервис-Ориентированная Архитектура – это…

Монолитная Архитектура – это…

Микросервисная Архитектура – это…

Сравнение Микросервисной и Монолитной Архитектур (таблица). Сложность разработки, сложность внесения изменений, заменяемость, развёртывание, сложность поддержки, отказоустойчивость, тестирование, масштабирование

Независимость микросервисов, противоречия: микросервисы должны общаться, но при этом быть как можно более независимыми).

Разнообразие используемых технологий для создания каждого микросервиса.

Возможности, доступные при использовании облачных технологий.

Балансировка нагрузки, возможность запускать несколько экземпляров(instance) каждого микросервиса.

Logging(журналирование) при использовании микросервисов, наличие централизованного хранилища логов.

# Глава 2 – Примеры использования и паттерны

Паттерн API Gateway и примеры (паттерн может быть использован во многих вариантах в зависимости от задач проекта)

Авторизация (в том числе в виде Gateway)

Паттерн API Decomposition – это…

Использование Message Queuing Сервисов (на примере RabbitMQ и BizTalk) для общения микросервисов, а также для инициации событий (web job).

Примеры проблем, возникающих при использовании Микросервисной Архитектуры.

# Глава 3 – Внутренняя архитектура каждого микросервиса.

Vertical Slice Архитектура, CQRS, возможность разделения ответственностей на Read/Write.

# Глава 4 – Проектирование архитектуры приложения.

Описание требований, задач, функционала.

Описание двух микросервисов, общающихся через RabbitMQ.

Графическое представление архитектуры приложения

Архитектура каждого микросервиса в отдельности.

# Глава 5 – Демонстрация работы проекта.

Скриншоты, демонстрация функций.

# Заключение

Короткий повтор преимуществ и недостаткой, а также описание проектов, для которых лучше подходит микросервисная или монолитная архитектура.

# Литература и документация:

\*ссылки, документация и книги\*